

REMARKS

This amendment responds to the Office Action mailed October 27, 2003. In the Office Action the Examiner:

- objected to the Abstract for being too long;
- rejected claims 22-51 under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-21 of Tock, et al., U.S. Patent No. 6,128,771 (hereinafter Tock); and
- rejected claims 22-51 under 35 U.S.C. 103(a) as being unpatentable over Hastings, U.S. Patent No. 5,193,180 (hereinafter Hastings) in view of Gupta, U.S. Patent No. 5,822,590 (hereinafter Gupta).

After entry of this amendment, the pending claims are: claims 22-51.

Abstract

Applicants have amended the Abstract such that its length is within the range of 50 to 150 words. Therefore, Applicants respectfully request that the objection be withdrawn.

Double patenting

Claims 22-51 are rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-21 of Tock, et al., U.S. Patent No. 6,128,771. Applicants respectfully traverse the rejection with a timely filed Terminal Disclaimer in compliance with 37 CFR 1.321(b) and (c), because U.S. Patent No. 6,128,771 is commonly owned with this application.

Rejections under 35 U.S.C. 103(a)

To reject claims in an application under 35 U.S.C. § 103, the Examiner bears the initial burden of establishing a prima facie case of obviousness. *In re Bell*, 26 USPQ2d 1529, 1530 (Fed. Cir. 1993). In order to establish prima facie obviousness, the prior art, alone or in combination, must teach or suggest each and every limitation of the rejected claims. See *In re Vaeck*, 20 USPQ2d 1438 (Fed. Cir. 1991); *In re Royka and Martin* 180 USPQ 580 (C.C.P.A. 1974); and *In re Wilson* 165 USPQ 494 (C.C.P.A. 1970). The teaching or suggestion to make the claimed invention, as well as the reasonable expectation of success,

must come from the prior art, not Applicants' disclosure. *In re Vaeck, Id.* For the reasons explained below, in the present instance, neither Hastings nor Gupta teach or suggest each limitation of the rejected claims. Therefore, Applicants respectfully traverse the rejections.

Claim 22 is directed to a method of generating object-oriented computer programs for accessing and updating objects stored in a persistent storage, e.g., a database. The first step of such a method is to receive an initial computer program. In particular, the initial program includes instructions for accessing persistent objects that store main memory copies of persistently stored objects. The method then scans the initial computer program to identify object accessing instructions and corresponding program locations at which additional instructions are to be added. The method also generates a revised computer program by modifying data structures of the persistent objects and adding object loading instructions to the initial computer program such that the added object loading instructions, during execution of the revised computer program, load respective ones of the persistent objects from persistent storage of the computer into the main memory. Support for claim 22 can be found in Tock (see on col. 7, lines 52-67, col. 10, lines 8-35, and Fig. 5).

In contrast, Hastings teaches away from the present invention or is inconsistent with the present invention in several ways. First, Hastings does not teach the limitation of modifying data structures of the persistent objects as recited in claim 22. Hastings teaches a method that "instruments" an object code by identifying instructions that access main memory and then adding instructions to (1) keep track of which memory locations have been utilized by the object code, and (2) to generate an error signal whenever the object code attempts to access a memory location that has not been utilized (e.g., which has not yet been allocated, or which does not store a legal value).

However, the added instructions of Hastings do not change any existing data structure of the original program. The method taught by Hastings is a generic one that can be applied to any type of object code as long as it has access to the main memory, be it object-oriented or not. Claim 22, as suggested by its preamble, is limited to object-oriented computer programs. Besides adding new instructions, Hastings also teaches how to modify offsets in certain preexisting instructions to compensate for the addition of the new instructions. But these offsets are adjusted only to reflect new positions of the preexisting instructions and data in the instrumented object code, and none of them are related to the data structure of any

existing object or data structure used by the original program. This aspect of the rewritten claims is supported by step 250 of Fig. 5 and the corresponding text of the specification.

Second, the focus of Hastings is any instruction that accesses main memory, while claim 22 identifies only instructions that access the persistent objects in the initial computer program. Hastings uses the word “object” only in the context of “object code” and does not distinguish between accessing persistent objects and accessing other data in memory. This distinction (between persistent objects and other types of data) is very important in the present invention, because it is only the instructions accessing and/or updating persistent objects that require special processing in the present invention. Hastings, therefore, does not “automatically identify ... object accessing instructions.”

Third, the instructions added by Hastings cause execution of the instrumented object code to generate an error signal when the object code attempts to access a memory location that does not contain any data. See col. 9, lines 15-55, with the error signal generation being indicated at lines 52 and 55. This is directly contrary to the teachings of the present invention, which teaches that the revised program loads an object into main memory the first time that the object is accessed by the revised program so as to avoid generating any error message.

Further, whatever similarities Gupta may have to the present invention, Gupta does not compensate for the above identified distinctions between the present invention, as claimed, and Hastings. For instance, Gupta does not teach or suggest the limitation of modifying data structures of the persistent objects. The only data structure explicitly discussed in Gupta is a global data structure, *dbtable*, that links the main memory and the hard disk of a computer. The concept of **global** in computer science, i.e., the data stored in the structure are shared by all variables of the same program, is directly contrary to the concept of **object-oriented** programming that emphasizes the privacy of one object's content from another. Since the combined teachings of Hastings and Gupta do not teach every limitation of claim 22, claim 22 and its dependent claims 23-25 are patentable over Hastings in view of Gupta.

Claims 26 and 28 recite methods that are similar to claim 22. Additionally, the method recited in claim 26 splits the identified program locations into two subsets, adding dirty object marking instructions at the first subset and object storing instructions at the second subset. Similarly, the method recited in claim 28 also splits the splits the identified

program locations into two subsets, adding at the first subset a first set of instructions that performs a first predefined task and at the second subset a second set of instructions that performs a second predefined task.

Neither Hastings nor Gupta have taught or even suggested a method that splits the identified program locations into two subsets and associates with each subset with one set of instructions, because there is no classification of added instructions in either one. Therefore, claims 26 and 28 as well as their respective dependent claims 27 and 29-31 are patentable over Hastings in view of Gupta.

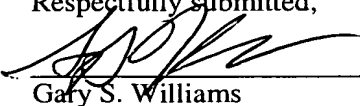
Claims 32, 36 and 38 are computer program product claims corresponding to claims 22, 26 and 28, respectively. Each of them explicitly recites a postprocessor procedure implementing its corresponding method claim. Therefore, claims 32, 36 and 38, and their respective dependent claims 33-35, 37 and 39-41, are patentable over Hastings in view of Gupta.

Claims 42, 46 and 48 are computer system claims corresponding to claims 22, 26 and 28, respectively. Each of them explicitly also recites a postprocessor procedure implementing its corresponding method claim. Therefore, claims 42, 46 and 48, and their respective dependent claims 43-45, 47 and 49-51, are patentable over Hastings in view of Gupta.

In light of the above amendments and remarks, Applicants respectfully request that the Examiner reconsider this application with a view towards allowance. The Examiner is invited to call the undersigned attorney at 650-849-7721, if a telephone call could help resolve any remaining items.

Date: January 26, 2004

Respectfully submitted,



Gary S. Williams
Morgan, Lewis & Bockius LLP
3300 Hillview Avenue
Palo Alto, California 94304
(650) 493-4935

31,066
(Reg. No.)